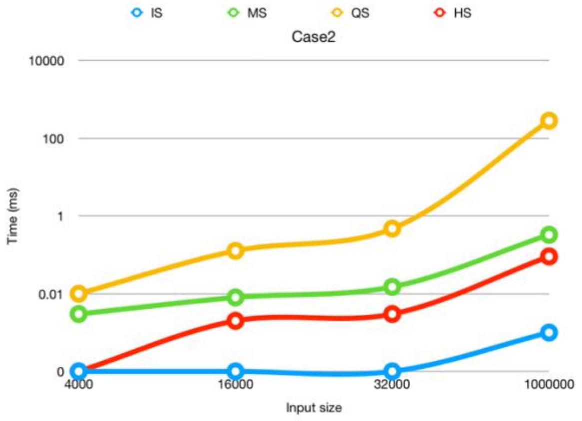
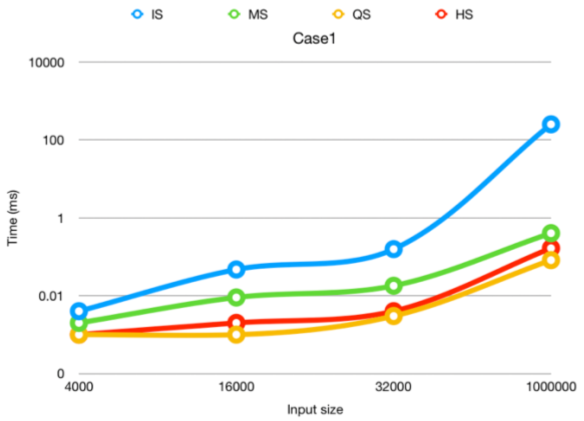
report

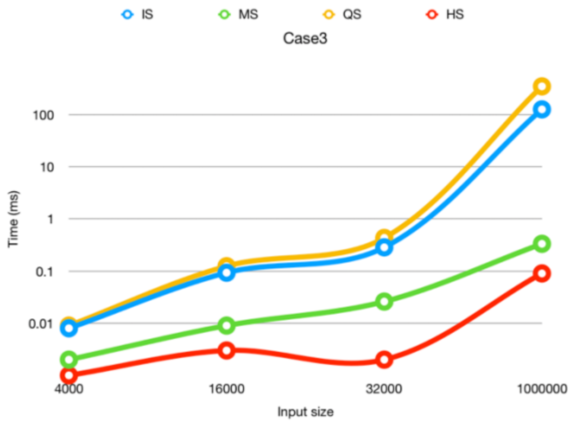
1. table

Input size	IS		MS		QS		HS	
	CPU time (ms)	Memory(KB)	CPU time (ms)	Memory(KB)	CPU time (ms)	Memory(KB)	CPU time (ms)	Memory(KB)
4000.case2	0	12500	3	12500	9.999	12616	0	12500
4000.case3	7.999	12500	2	12500	8.999	12520	1	12500
4000.case1	3.999	12500	2	12500	0.999	12500	1	12500
16000.case2	0	12648	7.999	12648	126.98	13316	2	12648
16000.case3	92.985	12648	8.999	12648	123.981	12944	3	12648
16000.case1	46.992	12648	8.999	12648	1	12648	1.999	12648
32000.case2	0	12648	14.997	13020	473.928	14052	2.999	12648
32000.case3	284.957	12648	25.996	13020	431.934	13316	2	12648
32000.case1	156.976	12648	17.997	13020	3	12648	4	12648
1000000.case2	1	18668	328.95	28336	282912	46164	90.987	18668
1000000.case3	125871	18668	332.949	28336	348028	34704	89.986	18668
1000000.case1	251702	18668	402.939	28336	81.987	18668	165.975	18668

2. figures







Analysis:

(1) We know that case2 is already sorted and case3 is in reversed order.

(2) Case2: IS performs best and QS performs worst. See 3(1),(3).

(3) Case1: since case1 is not a special case, the differences in performance between those tools are smaller than other two cases. In case1, QS have the best performance, while IS

performs worst.

(4) Case3: MS have good performance with time complexity O(nlogn). QS and IS have nearly the same performance with time complexity O(n^2).

(5) The difference in running time between those different cases: IS is the biggest and MS is the smallest.

(6) MS and HS perform almost the same in each case.

3. Implementation and findings

(1) IS: use vector to implement. Case2 is the best-case scenario, which takes linear time. Case3 is the worst-case scenario, which takes O(n^2) time.

(2) MS: use vector to implement. In each iteration, the data be cut half and the split is nearly 1:1, so the running time of each case is almost the same. Time complexity is O(nlogn).

(3) QS: use vector to implement. Choosing the last element as pivot in each iteration, case2(already sorted) and case3(reversed) become the worst case, since there is a (n-1: 0) split in each partition. In this case, time complexity is O(n^2). Perhaps selecting the pivot randomly will have better performance in these cases.

(4) HS: to convert input to a heap and heapify recursively. The time complexity is O(nlogn).

(5) For these four sort tools, IS, QS and HS are in-place while MS is not. When implementing MS, I have used another vector to merge the two sub-vector (left and right).

(6) MS and HP have rather more stable performance in these cases.